intel®

ARTICLE
REPRINT

AR-40

July 1977

Slave microcomputer lightens main microprocessor load

by Don Phillips and Allen Goodman, *Intel Corp., Santa Clara, Calif.*

Electronics/July 7, 1977

Slave microcomputer lightens main microprocessor load

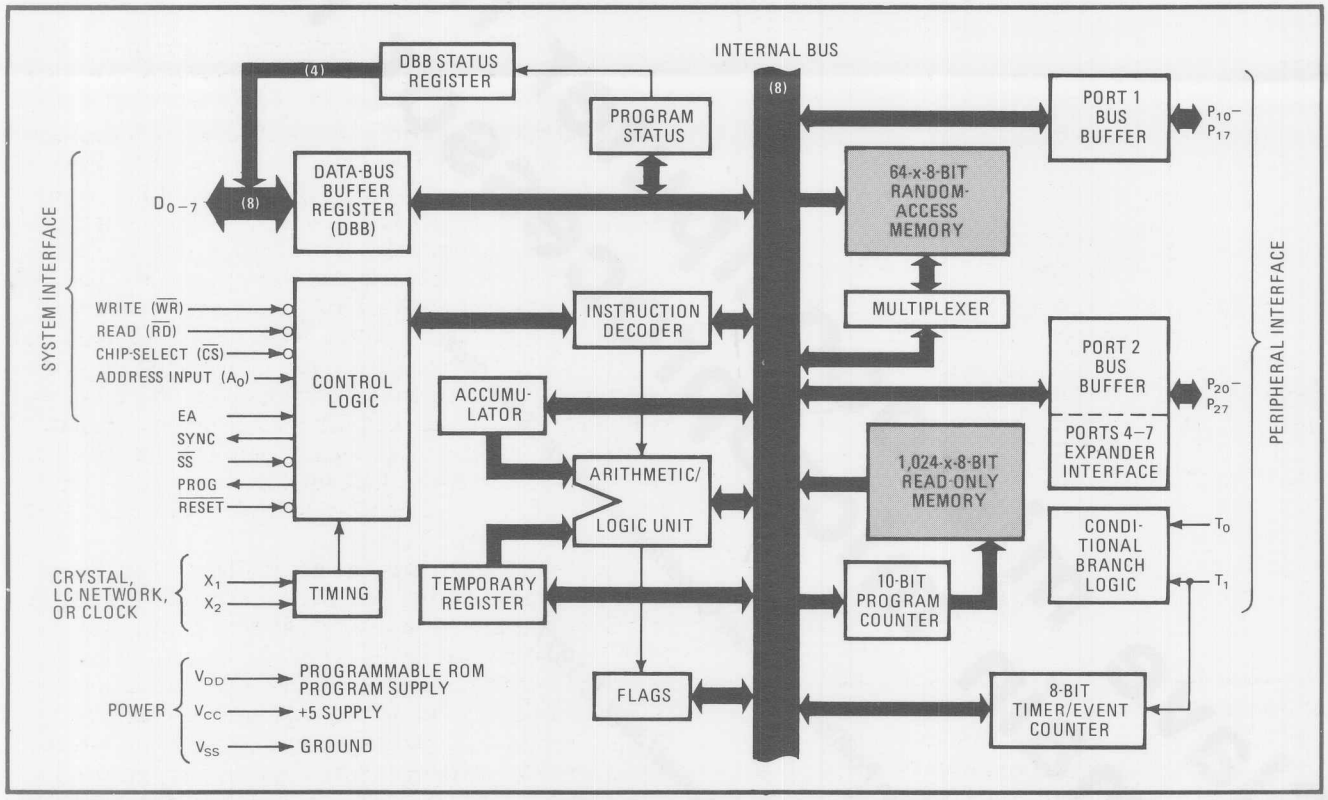
by Don Phillips and Allen Goodman, *Intel Corp., Santa Clara, Calif.*

☐ Peripheral devices for microprocessors are growing in number and complexity to the point where they are taxing the processor's time and memory. Nor do simple interface adapters that contain no intelligence of their own lighten the burden of managing such peripheral equipment as floppy disks, cathode-ray-tube displays, and keyboards. What can save the day for the central processing unit is a new class of peripheral controllers: intelligent microcomputer-based universal peripheral interface chips.

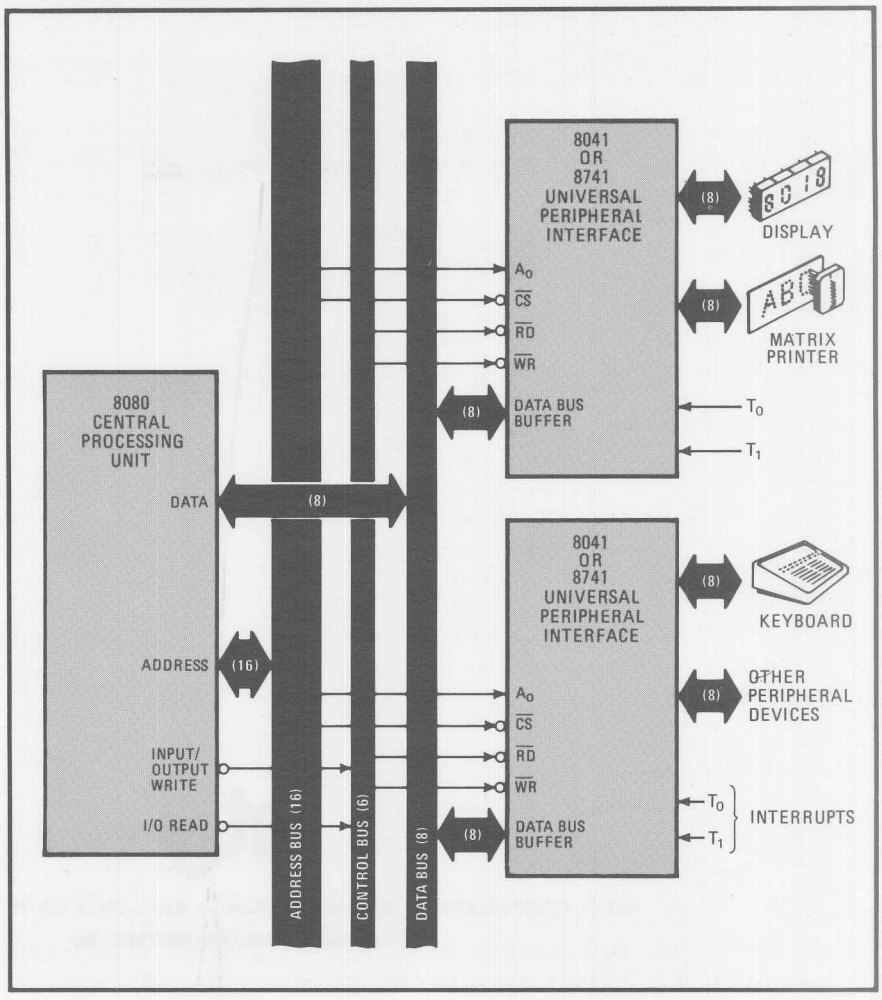
In essence, what the UPI microcomputer does is act as a slave processor to the main-system CPU. With a built-in processor and memory, it greatly eases the handling of real-time tasks such as controlling printers, encoding keyboards, and multiplexing displays. In fact, entire control algorithms can be programmed locally in the slave processor, instead of taxing the limited memory space and execution time of the main system. Moreover, the device substantially increases the overall efficiency of a system, since two processors—the central CPU and the slave UPI device—are working in parallel.

A peripheral controller

In operation, the UPI microcomputer acts as a peripheral controller rather than just an interface adapter. Its architecture, detailed in Fig. 1, is similar to the recently introduced 8048 one-chip microcomputer: it has an 8-bit CPU, 64 bytes of random-access memory, 1,024 bytes of read-only memory, a timer/counter, and 18 input/output lines. In fact, the device executes the same basic set of instructions as does the 8048, except for special tailoring of data-bus operations to better suit control applications. The difference is that the new peripheral-controlling microcomputer is designed to function as a



1. Smart interface. With an 8-bit CPU, 64 bytes of RAM, and 1,024 words of ROM or erasable PROM, the universal peripheral interface chip is an intelligent peripheral controller rather than a simple interface adapter. The architecture of the chip is similar to that of the 8048 microcomputer. It uses nearly the same instruction set, save for slight variations that improve data-bus operations.



2. Slaves. The microcomputer-based universal peripheral interface chips—the 8741 with erasable PROM and the 8041 with mask-programmed ROM—are connected as slave processors to a main processor (here an 8080 CPU) to take over its I/O chores.

slave processor to the main-system processor.

The chip is the first microcomputer made specifically for a multiprocessor environment in which a master processor sends information to one or more slave processors that in turn control peripheral devices. To accommodate a variety of master processor types, including the 8080, the enhanced 8085, and other 8-bit processors, the chip has bus interface registers that work directly with the central processor's data bus.

Two peripheral controllers are available: the 8741 and the 8041, identical except in one respect. The 8741 has an ultraviolet-erasable, electrically programmable ROM plus the special capability of running through a program a single step at a time. It is designed for low-volume applications requiring program development, as in prototype testing and custom interfacing. The 8041 has a conventional mask-programmable ROM and features a low-power standby mode. It is intended to replace the 8741 once a system design has been set. The 8741/8041 connections for a master-slave arrangement are shown in the block diagram of Fig. 2.

The master processor and the peripheral controller communicate through an asynchronous data-bus buffer register on the UPI. Data and commands are received from the master processor through the DBB, and status and data information are returned through it to the master. The controller sends status information to the main processor from a 4-bit status register that uses four of the buffer register's eight lines.

The configuration of the DBB and status registers is shown in Fig. 3. The master processor controls data transfer to the UPI by four input lines: the address-input signal specifies whether a command or a data word is being sent; the chip-select line is an enable input that permits communication with the interface, and the read and write lines are used to stroke output and input data, respectively. The master processor uses these signals to direct the exchange of information through the DBB register, which serves as temporary storage for commands and data flowing between master and slave processors.

The four flags

The status register comprises four flags that direct the handshaking between the master and slave processors. The first is a general-purpose flag, which is set by programming in the 8041/8741 and used to prevent contention over the DBB register between master and slave processors. Another is the command/data flag that, when set, indicates that command information is being transferred. The input-buffer-full flag is set whenever the DBB register is loaded with a word from the main processor, and the output-buffer-full flag is set when the UPI loads its DBB register.

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 (408) 246-7501
Printed in U.S.A./A231/0877/10K BL

Protocol for the interface begins with the master processor writing an 8-bit character into the buffer register. This sets the IBF flag, signaling the peripheral controller with an internal interrupt. The UPI can then transfer the 8-bit data byte to its accumulator at any time under software control, which clears the IBF flag.

In transferring data in the other direction—from slave to master—the peripheral chip loads the DBB register while automatically setting the OBF flag. The master processor can then read the status register to determine that the OBF flag is set and can proceed to take in data from the buffer register, at the same time clearing the flag in preparation for the arrival of more data.

Transfer of data within the peripheral controller is asynchronous to external processor timing. The chip can thus effectively control peripheral devices while data transfers go on unhindered. Moreover, the DBB register isolates peripheral control tasks from the main processor. Task isolation is desirable in that it eases software development and debugging within a given system (by modularizing functions). In addition, it is certain to enhance data throughput, since two microprocessors are running concurrently.

Optimized for control

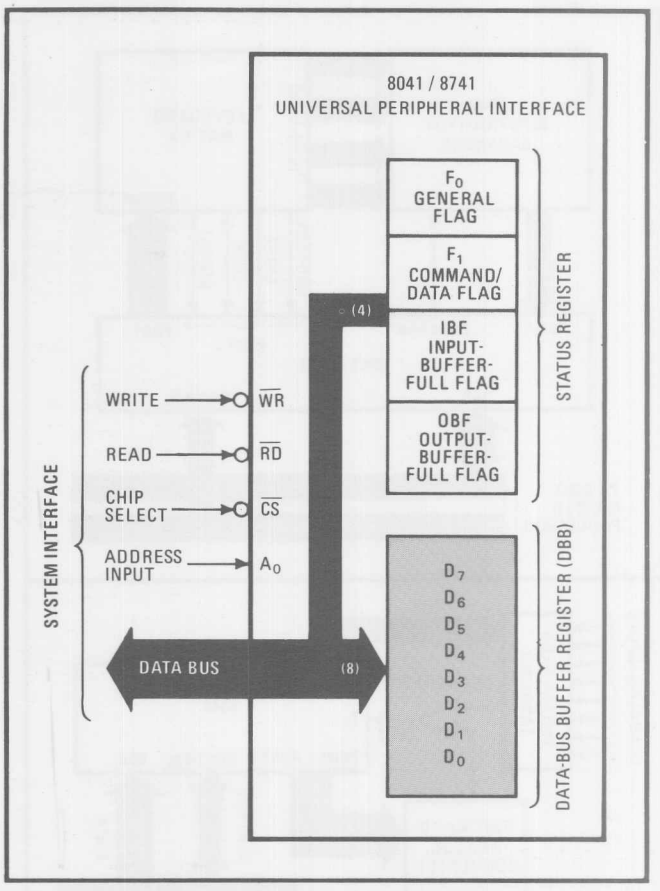
The CPU and instruction set of the 8041/8741 are designed to efficiently handle the single-bit operations required in most control applications, including I/O operations and data-bit manipulation. Two 8-bit-wide ports, compatible with transistor-transistor logic, are provided on the chip. (Sixteen additional lines may be had with the addition of an 8243 I/O expander chip, which takes up half the lines of I/O port 2.) Two inputs to the peripheral controller are provided that may be tested with conditional branch instructions in UPI software. Any port line can be set or cleared individually under software control, and any line can function as either input or output, irrespective of remaining lines.

The timer/event-counter included on the peripheral controller can be preset, read, started, or stopped under software control. In the timing mode, an internal oscillator can be set by a crystal or an LC network. In the event-counter mode, the T_1 input may be used to count switch closures or tachometer pulses, directing program flow accordingly. If the counter has been preset, a flag is available that indicates overflow, and it can signal the master processor.

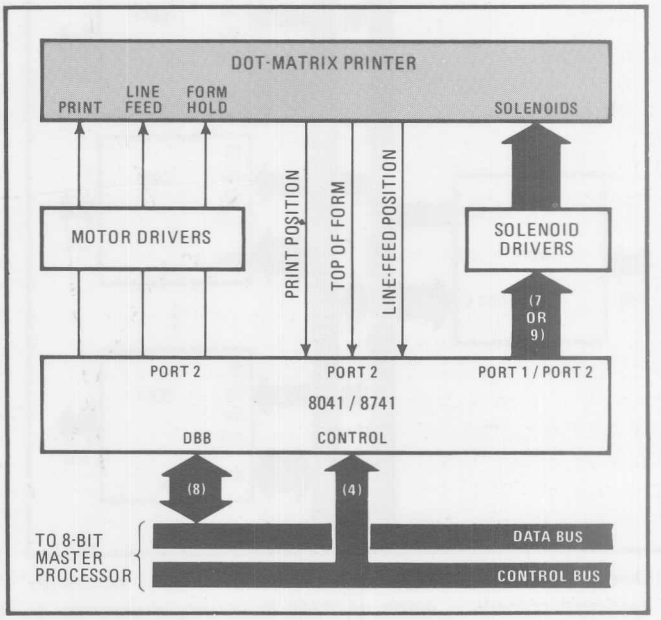
The 1,024 bytes of on-chip ROM are sufficient for most dedicated programming applications. Typically, keyboard encoding or printer control requires 500 to 700 8-bit bytes, and therefore ample program space is available for additional functions.

Of the 64 locations in the on-chip RAM, there are two 8-byte register banks, an eight-level program-counter stack, and 32 bytes of user RAM. The dual 8-byte register banks allow fast response to interrupts such as the IBF flag or time overflow. The stack also provides convenient handling of subroutine cells and storage of other data.

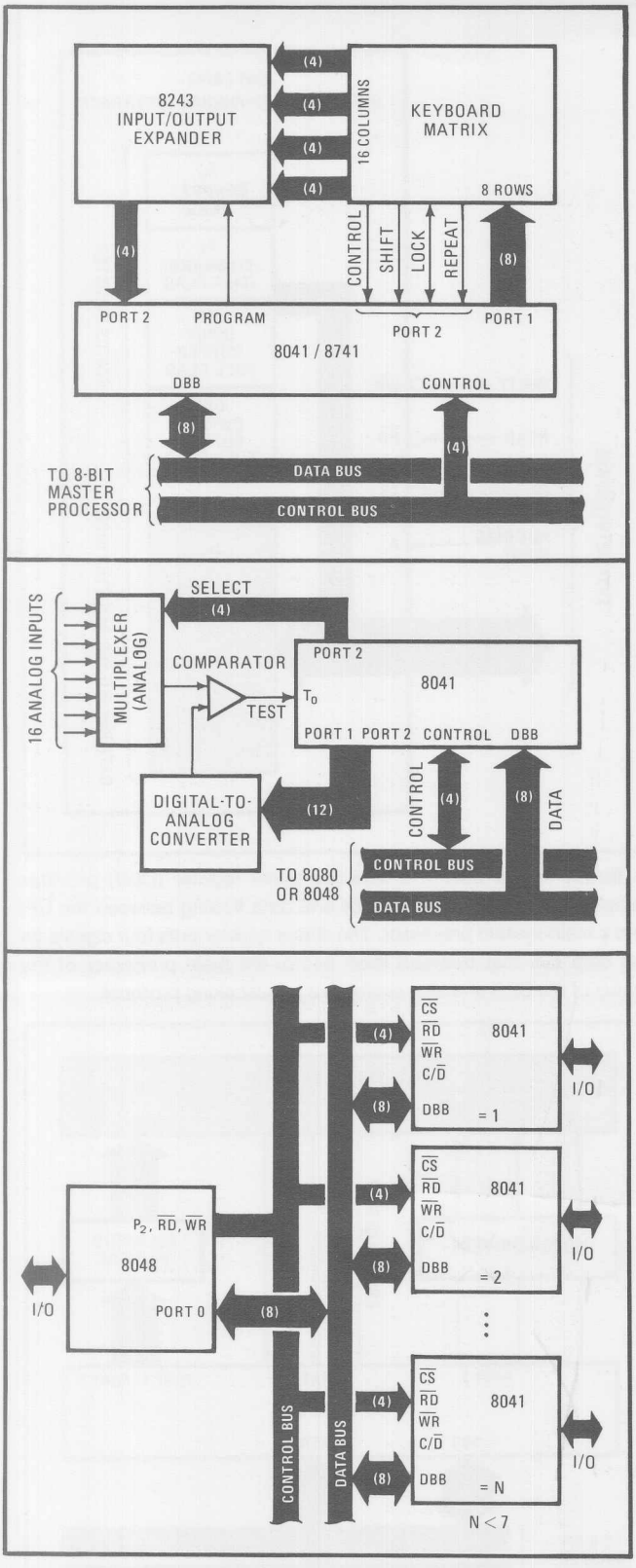
The thrust of the peripheral-controller chip is in its isolation of peripheral tasks from the main processor. Since its job is specifically for control, the main



3. Buffer to the bus. The data-bus buffer register (DBB) provides temporary storage for commands and data flowing between the UPI and a main-system processor. The status register puts four signals on the data bus that between them inform the main processor of the status of the DBB and also establish a handshaking protocol.



4. Printer control. Memory in the 8741/8041 allows the device to buffer as many as 40 characters to be printed. The main-system processor can transfer a block of data at this speed and then continue with other tasks while the UPI's bidirectional I/O ports monitor and control sequential character printing.



5. Using the UPI. Typical applications of the 8741/8041 include (a) a keyboard scanner in which an 8243 input/output expander is added to permit the encoding of as many as 128 keys, (b) a process-control subsystem implemented with an analog multiplexer and a digital-to-analog converter, and (c) a generalized distributed processing system that employs up to seven of the devices as slave processors, connecting them to a single 8048 microcomputer.

processor can therefore be left to down-load commands and transfer data, while the UPI works in real time.

One application might be the controlling of a printer peripheral to an 8080 system, as shown in Fig. 4. The entire real-time control portion of the task can be handled by the peripheral controller. With its built-in timer, it easily handles timing functions like character spacing, print position, and line feed. The UPI has ample I/O ports for a 40-column dot-matrix printer.

In this printer application, the DBB register allows for standardization of data transfer to and from the 8080-based main processing system. To do this, one typical format might be for the main processor to send a start command followed by a full line of 40 ASCII characters. The peripheral controller would then store the characters under program control in a portion of the RAM and begin execution of the printing as soon as the print head and line feed were in the proper position. In the meantime, the main processor returns to other tasks. The ROM in the 8041/8741 can be used to convert the ASCII code to dot-matrix or other formats.

In printer applications, standardization is the key feature offered by the slave peripheral controller. Without any changes in the 8080-based main processing system, the UPI can be programmed to handle any printer mechanisms—dot matrix, drum, spherical head, and so on. In this way, a designer can easily upgrade the peripherals in his system with a minimum of change in the master-processor software.

A keyboard application

Figure 5a illustrates an application in which the new chip plus an 8243 I/O expander provide a compact system for scanning and encoding as many as 128 keys from a terminal keyboard. N-key rollover and debounce are implemented by using the on-chip RAM to keep a copy of the key status after each scan. When a key closure is detected, the 8041 uses a ROM look-up table to generate the appropriate ASCII code for transfer to the master processor. As many as 16 characters can be stored and transferred in a block to the master processor.

The analog process-control subsystem illustrated in Fig. 5b can be implemented using an analog multiplexer and digital-to-analog converter along with the 8041. In this configuration, the peripheral controller can monitor and digitize eight analog inputs, perform linearization (using equations or ROM look-up tables), check for limits and zero offsets, and receive control information that could determine new limits.

Figure 5c illustrates a generalized distributed-processing system using as many as seven 8041s as slave processors connected to a single 8048 master processor. Port 2 of the 8048 provides seven chip-select lines to the peripheral controllers plus the command/data control function. This low-cost, low-speed multiprocessor configuration has many advantages over a single high-speed processor. The peripheral controllers are designed especially for control or interface applications, and each can be programmed to handle a single isolated task. This modular approach allows easy development and debugging of the system. □